

Scala Scripting

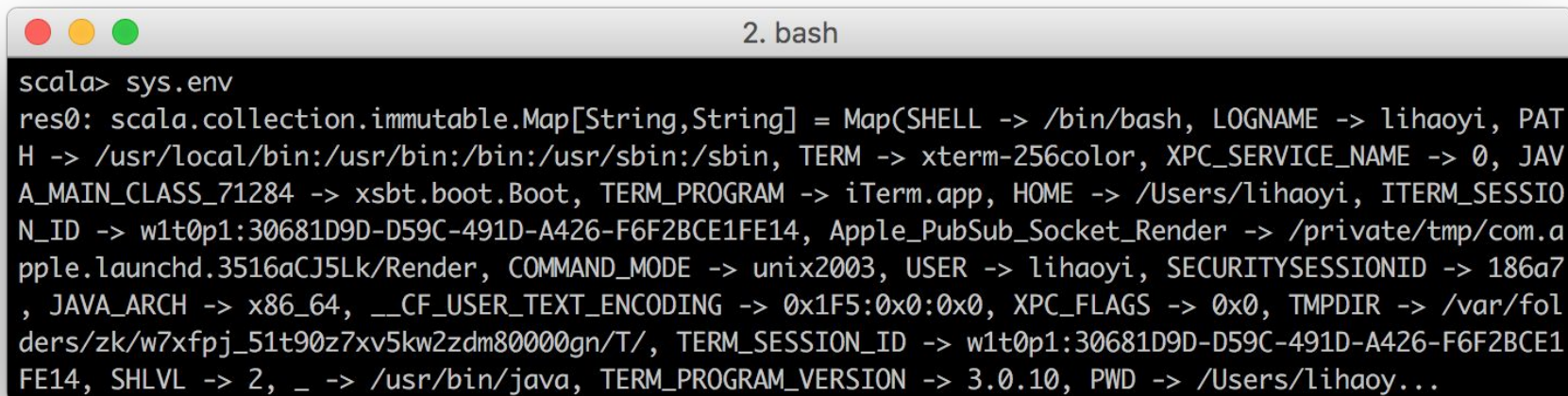
Scala By the Bay, San Francisco, 12 Nov 2016

Scala has a code-size gap

Scala REPL: lightweight, works great for < 1 line programs, more is problematic

Scala has a code-size gap

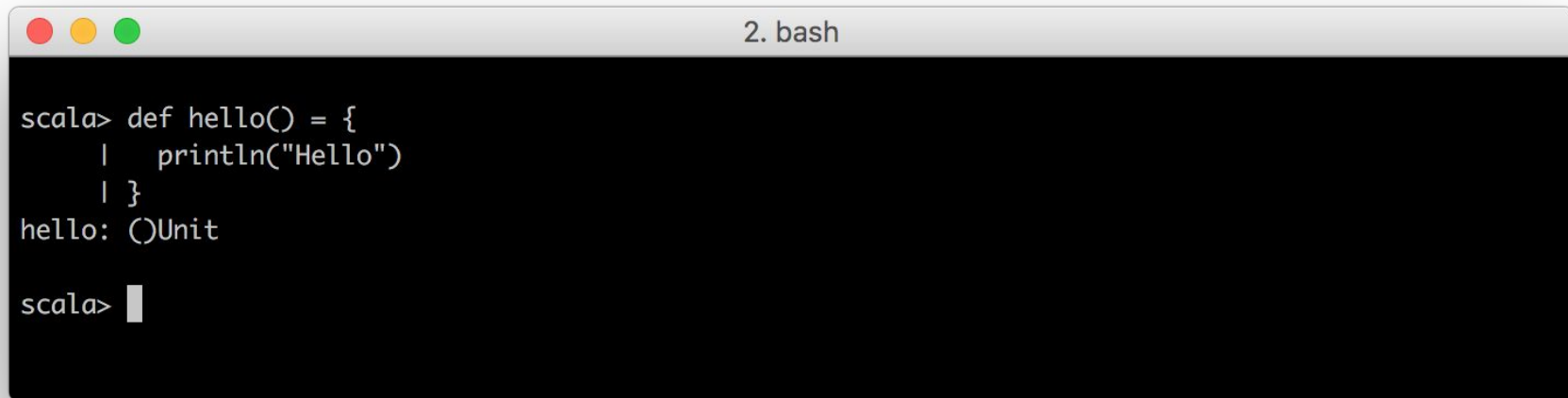
Scala REPL: lightweight, works great for < 1 line programs, more is problematic



```
2. bash
scala> sys.env
res0: scala.collection.immutable.Map[String,String] = Map(SHELL -> /bin/bash, LOGNAME -> lihaoyi, PATH -> /usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin, TERM -> xterm-256color, XPC_SERVICE_NAME -> 0, JAVA_MAIN_CLASS_71284 -> xsbt.boot.Boot, TERM_PROGRAM -> iTerm.app, HOME -> /Users/lihaoyi, ITerm_SESSION_ID -> w1t0p1:30681D9D-D59C-491D-A426-F6F2BCE1FE14, Apple_PubSub_Socket_Render -> /private/tmp/com.apple.launchd.3516aCJ5Lk/Render, COMMAND_MODE -> unix2003, USER -> lihaoyi, SECURITYSESSIONID -> 186a7, JAVA_ARCH -> x86_64, __CF_USER_TEXT_ENCODING -> 0x1F5:0x0:0x0, XPC_FLAGS -> 0x0, TMPDIR -> /var/folders/zk/w7xfpj_51t90z7xv5kw2zdm80000gn/T/, TERM_SESSION_ID -> w1t0p1:30681D9D-D59C-491D-A426-F6F2BCE1FE14, SHLVL -> 2, _ -> /usr/bin/java, TERM_PROGRAM_VERSION -> 3.0.10, PWD -> /Users/lihaoyi...
```

Scala has a code-size gap

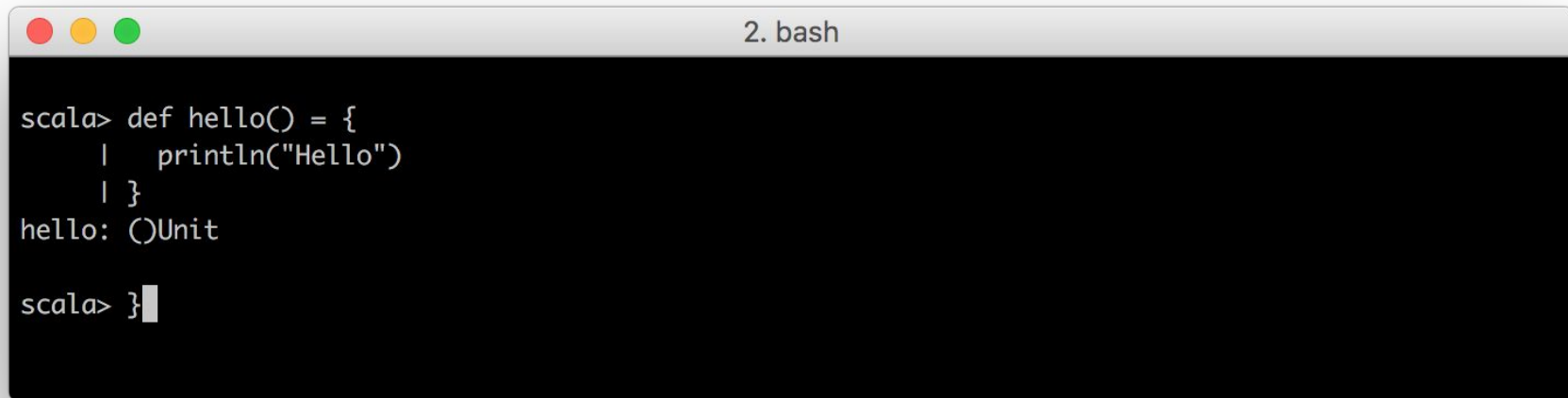
Scala REPL: lightweight, works great for < 1 line programs, more is problematic

A screenshot of a terminal window titled "2. bash". The terminal has a black background with white text. The text shows a Scala REPL session where a function named "hello" is defined. The function body consists of a single line: "println('Hello')". The REPL returns the type signature "hello: ()Unit". The prompt "scala>" is followed by a vertical bar cursor.

```
scala> def hello() = {  
  |   println("Hello")  
  | }  
hello: ()Unit  
  
scala> |
```

Scala has a code-size gap

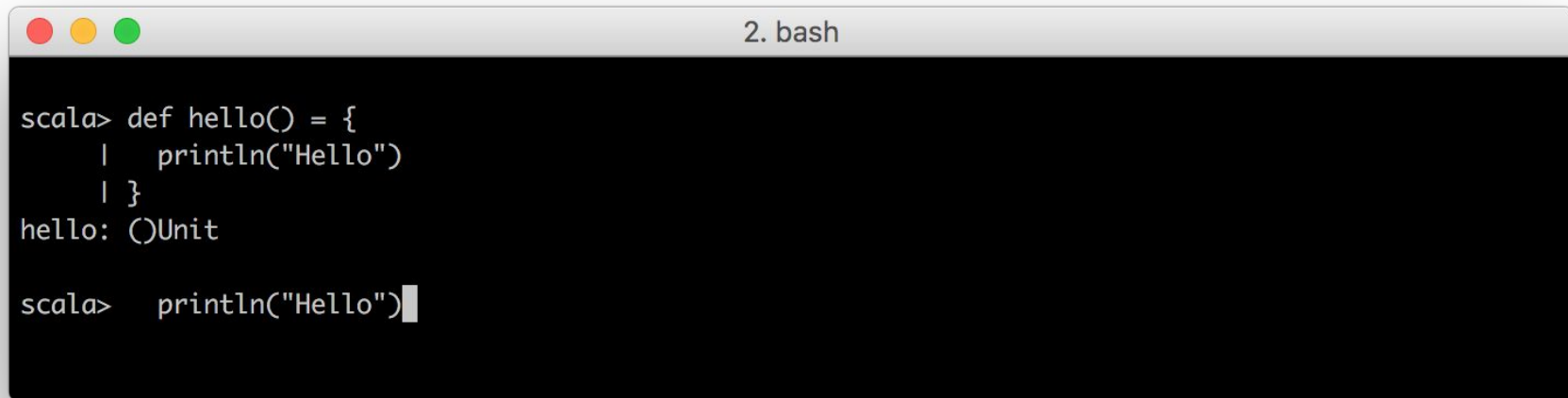
Scala REPL: lightweight, works great for < 1 line programs, more is problematic

A screenshot of a terminal window titled "2. bash". The terminal has a black background with white text. The text shows a Scala REPL session where a function named "hello" is defined with a single line of code: "println('Hello')". The REPL returns the type signature "hello: ()Unit". The user then enters a closing brace "}" and the cursor is positioned at the end of the line.

```
scala> def hello() = {  
  |   println("Hello")  
  | }  
hello: ()Unit  
  
scala> }|
```

Scala has a code-size gap

Scala REPL: lightweight, works great for < 1 line programs, more is problematic

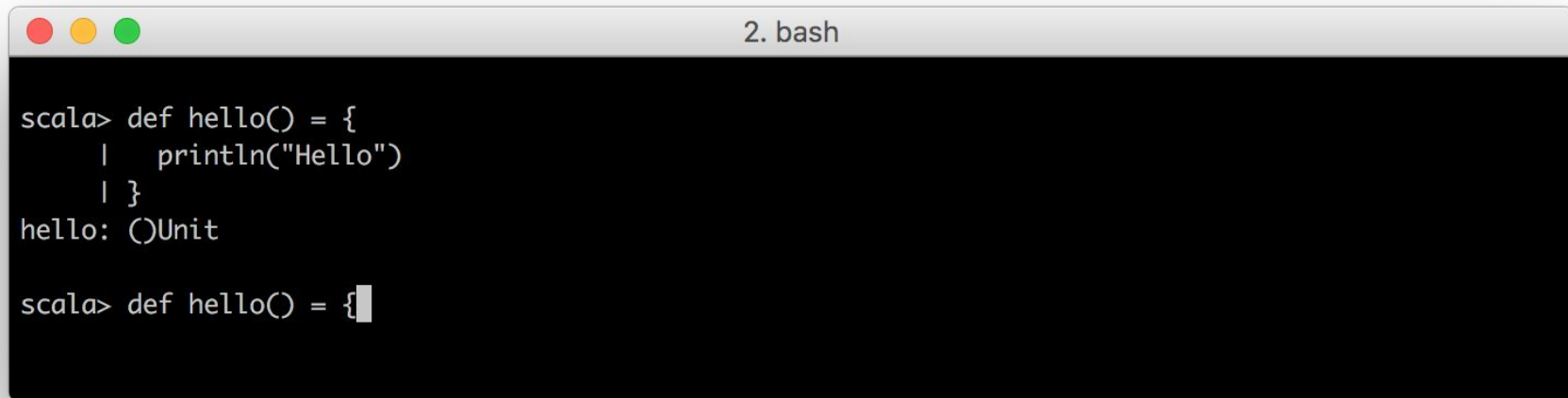
A screenshot of a terminal window with a dark background and light text. The window title bar shows three colored window control buttons (red, yellow, green) on the left and the text "2. bash" on the right. The terminal content shows a Scala REPL session where a function named 'hello' is defined and then called. The function definition is on three lines: 'scala> def hello() = {', ' | println("Hello")', and ' | }'. The return type 'hello: ()Unit' is displayed on the next line. The final line shows 'scala> println("Hello")' with a white cursor at the end of the line.

```
scala> def hello() = {
  |   println("Hello")
  | }
hello: ()Unit

scala> println("Hello")
```

Scala has a code-size gap

Scala REPL: lightweight, works great for < 1 line programs, more is problematic

A screenshot of a terminal window with a title bar containing three colored circles (red, yellow, green) and the text "2. bash". The terminal content shows a Scala REPL session. The first command is "scala> def hello() = {" followed by an indented line " | println(\"Hello\")" and another indented line " | }". The next line is "hello: ()Unit". The second command is "scala> def hello() = {" followed by a cursor. The terminal background is black and the text is white.

```
scala> def hello() = {  
  |   println("Hello")  
  | }  
hello: ()Unit  
  
scala> def hello() = {█
```

Scala has a code-size gap

SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts

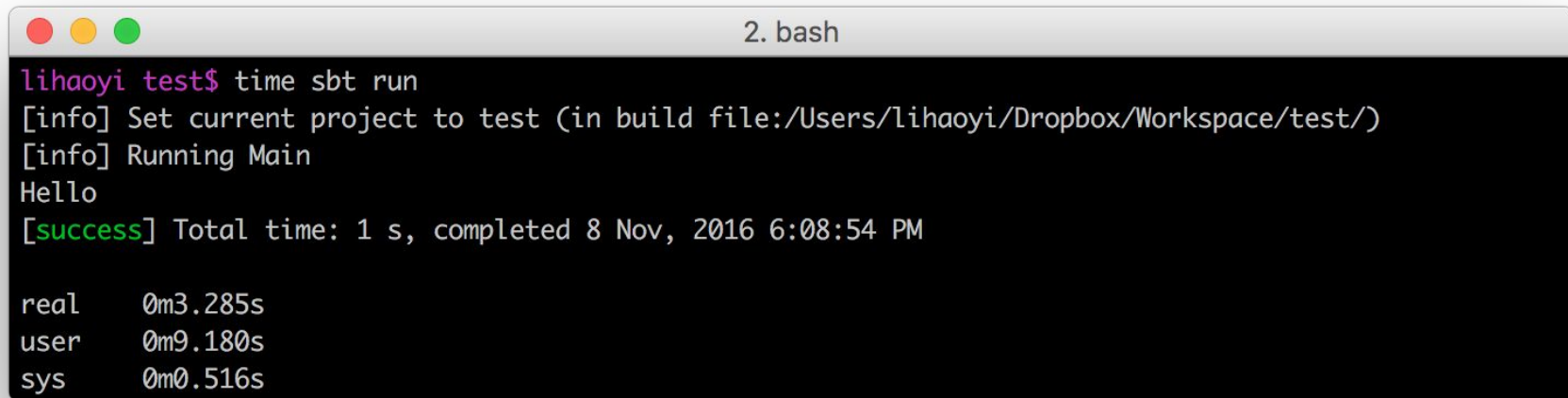
Scala has a code-size gap

```
object Main{  
    def main(args: Array[String]): Unit = {  
        println("Hello")  
    }  
}
```

SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts

Scala has a code-size gap

SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts

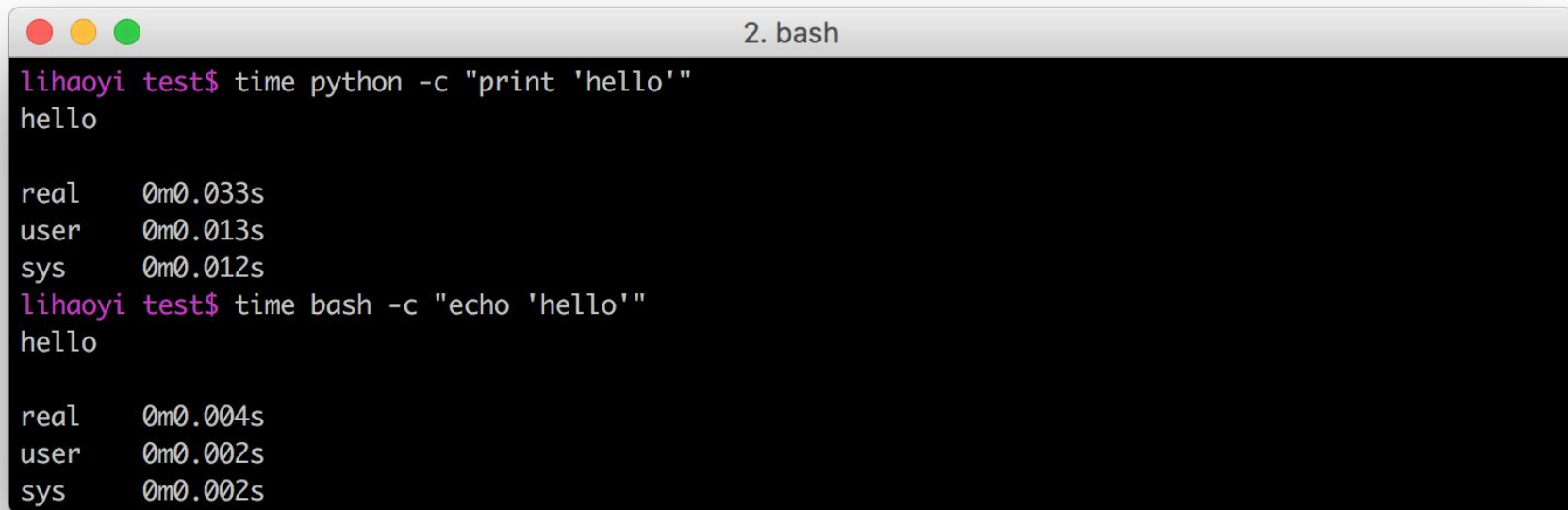
A terminal window titled "2. bash" with a dark background and light text. The window shows the execution of the SBT command 'time sbt run'. The output includes status messages from SBT, the program output 'Hello', and a summary of execution time. The terminal text is as follows:

```
lihaoyi test$ time sbt run
[info] Set current project to test (in build file:/Users/lihaoyi/Dropbox/Workspace/test/)
[info] Running Main
Hello
[success] Total time: 1 s, completed 8 Nov, 2016 6:08:54 PM

real    0m3.285s
user    0m9.180s
sys     0m0.516s
```

Scala has a code-size gap

SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts



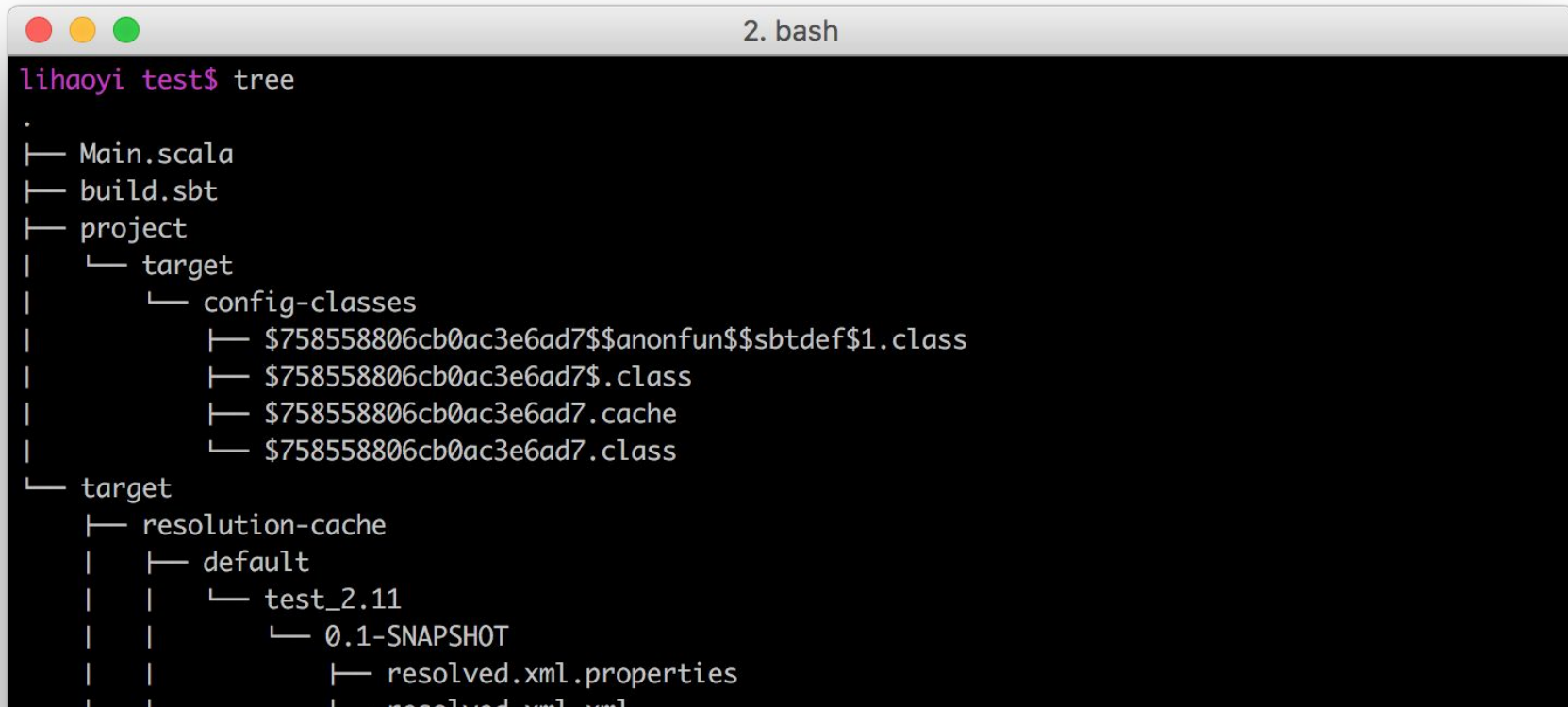
```
2. bash
lihaoyi test$ time python -c "print 'hello'"
hello

real    0m0.033s
user    0m0.013s
sys     0m0.012s
lihaoyi test$ time bash -c "echo 'hello'"
hello

real    0m0.004s
user    0m0.002s
sys     0m0.002s
```

Scala has a code-size gap

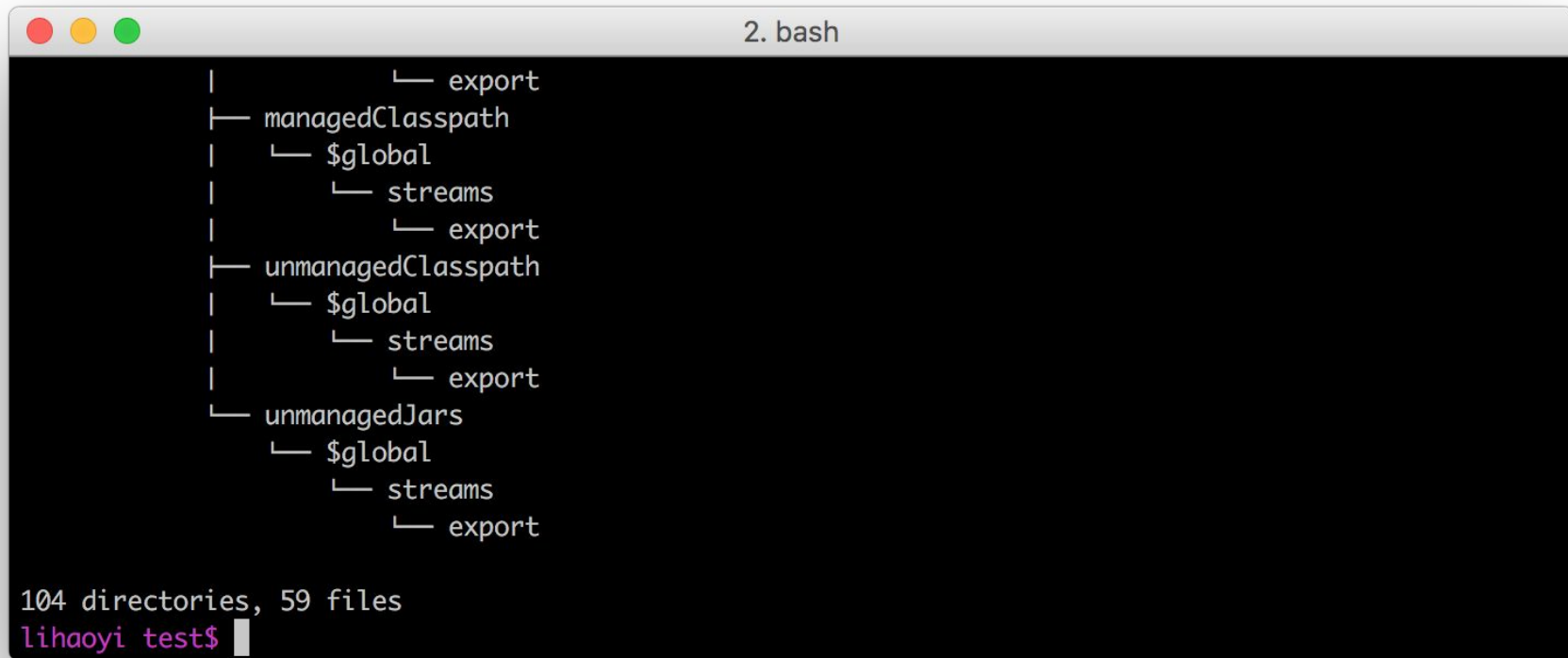
SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts



```
2. bash
lihaoyi test$ tree
.
├─ Main.scala
├─ build.sbt
├─ project
│   └─ target
│       └─ config-classes
│           ├── $758558806cb0ac3e6ad7$$anonfun$$sbtdef$1.class
│           ├── $758558806cb0ac3e6ad7$.class
│           ├── $758558806cb0ac3e6ad7.cache
│           └─ $758558806cb0ac3e6ad7.class
└─ target
    ├── resolution-cache
    │   └─ default
    │       └─ test_2.11
    │           └─ 0.1-SNAPSHOT
    │               └─ resolved.xml.properties
    └─ resolved.xml.xml
```

Scala has a code-size gap

SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts



```
2. bash
├── export
├── managedClasspath
│   ├── $global
│   ├── streams
│   └── export
├── unmanagedClasspath
│   ├── $global
│   ├── streams
│   └── export
├── unmanagedJars
│   ├── $global
│   ├── streams
│   └── export
└──
```

104 directories, 59 files
lihaoyi test\$

Scala has a code-size gap

Scala REPL: lightweight, works great for < 1 line programs, more is problematic

SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts

Scala has a code-size gap

Scala REPL: lightweight, works great for < 1 line programs, more is problematic



?

SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts

Scala has a code-size gap

Scala REPL: lightweight, works great for < 1 line programs, more is problematic



**Ammonite Scala
REPL/Scripts**

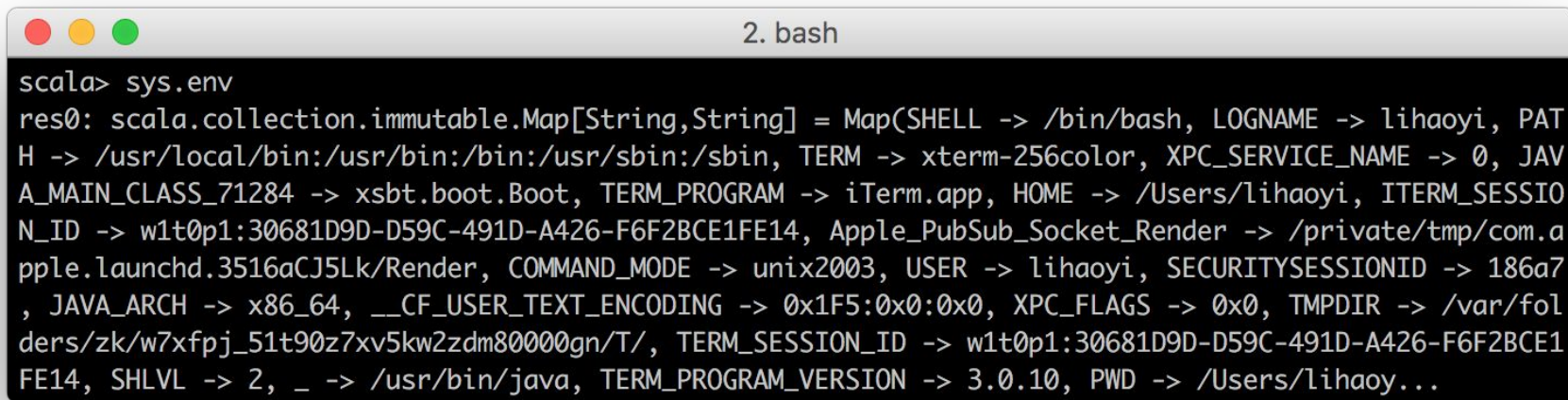
SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts

Ammonite: Scala Scripting

lihaoyi.github.io/Ammonite

Ammonite-REPL

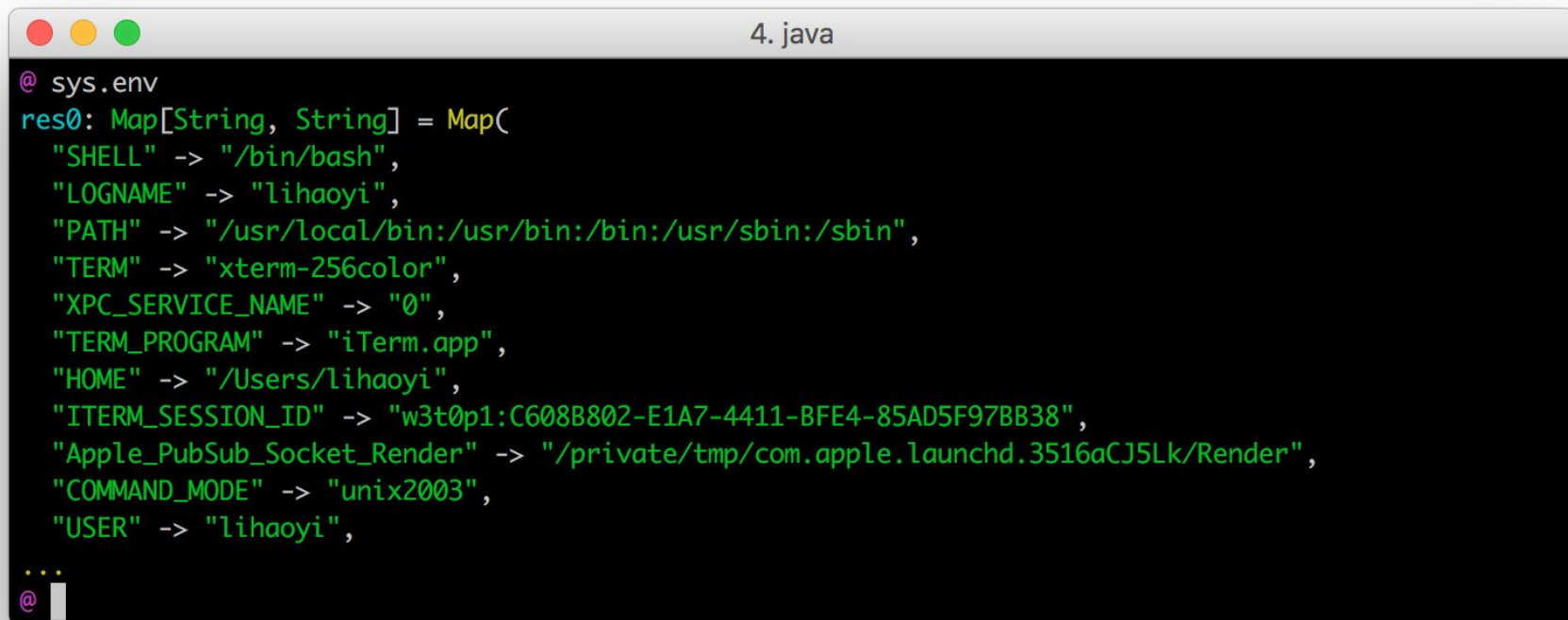
Like the Scala REPL,

A terminal window titled "2. bash" with a dark background and light text. The prompt is "scala> sys.env". The output is a long line of Scala code representing environment variables: "res0: scala.collection.immutable.Map[String,String] = Map(SHELL -> /bin/bash, LOGNAME -> lihaoyi, PATH -> /usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin, TERM -> xterm-256color, XPC_SERVICE_NAME -> 0, JAVA_MAIN_CLASS_71284 -> xsbt.boot.Boot, TERM_PROGRAM -> iTerm.app, HOME -> /Users/lihaoyi, ITerm_SESSION_ID -> w1t0p1:30681D9D-D59C-491D-A426-F6F2BCE1FE14, Apple_PubSub_Socket_Render -> /private/tmp/com.apple.launchd.3516aCJ5Lk/Render, COMMAND_MODE -> unix2003, USER -> lihaoyi, SECURITYSESSIONID -> 186a7, JAVA_ARCH -> x86_64, __CF_USER_TEXT_ENCODING -> 0x1F5:0x0:0x0, XPC_FLAGS -> 0x0, TMPDIR -> /var/folders/zk/w7xfpj_51t90z7xv5kw2zdm80000gn/T/, TERM_SESSION_ID -> w1t0p1:30681D9D-D59C-491D-A426-F6F2BCE1FE14, SHLVL -> 2, _ -> /usr/bin/java, TERM_PROGRAM_VERSION -> 3.0.10, PWD -> /Users/lihaoyi...".

```
scala> sys.env
res0: scala.collection.immutable.Map[String,String] = Map(SHELL -> /bin/bash, LOGNAME -> lihaoyi, PATH -> /usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin, TERM -> xterm-256color, XPC_SERVICE_NAME -> 0, JAVA_MAIN_CLASS_71284 -> xsbt.boot.Boot, TERM_PROGRAM -> iTerm.app, HOME -> /Users/lihaoyi, ITerm_SESSION_ID -> w1t0p1:30681D9D-D59C-491D-A426-F6F2BCE1FE14, Apple_PubSub_Socket_Render -> /private/tmp/com.apple.launchd.3516aCJ5Lk/Render, COMMAND_MODE -> unix2003, USER -> lihaoyi, SECURITYSESSIONID -> 186a7, JAVA_ARCH -> x86_64, __CF_USER_TEXT_ENCODING -> 0x1F5:0x0:0x0, XPC_FLAGS -> 0x0, TMPDIR -> /var/folders/zk/w7xfpj_51t90z7xv5kw2zdm80000gn/T/, TERM_SESSION_ID -> w1t0p1:30681D9D-D59C-491D-A426-F6F2BCE1FE14, SHLVL -> 2, _ -> /usr/bin/java, TERM_PROGRAM_VERSION -> 3.0.10, PWD -> /Users/lihaoyi...
```

Ammonite-REPL

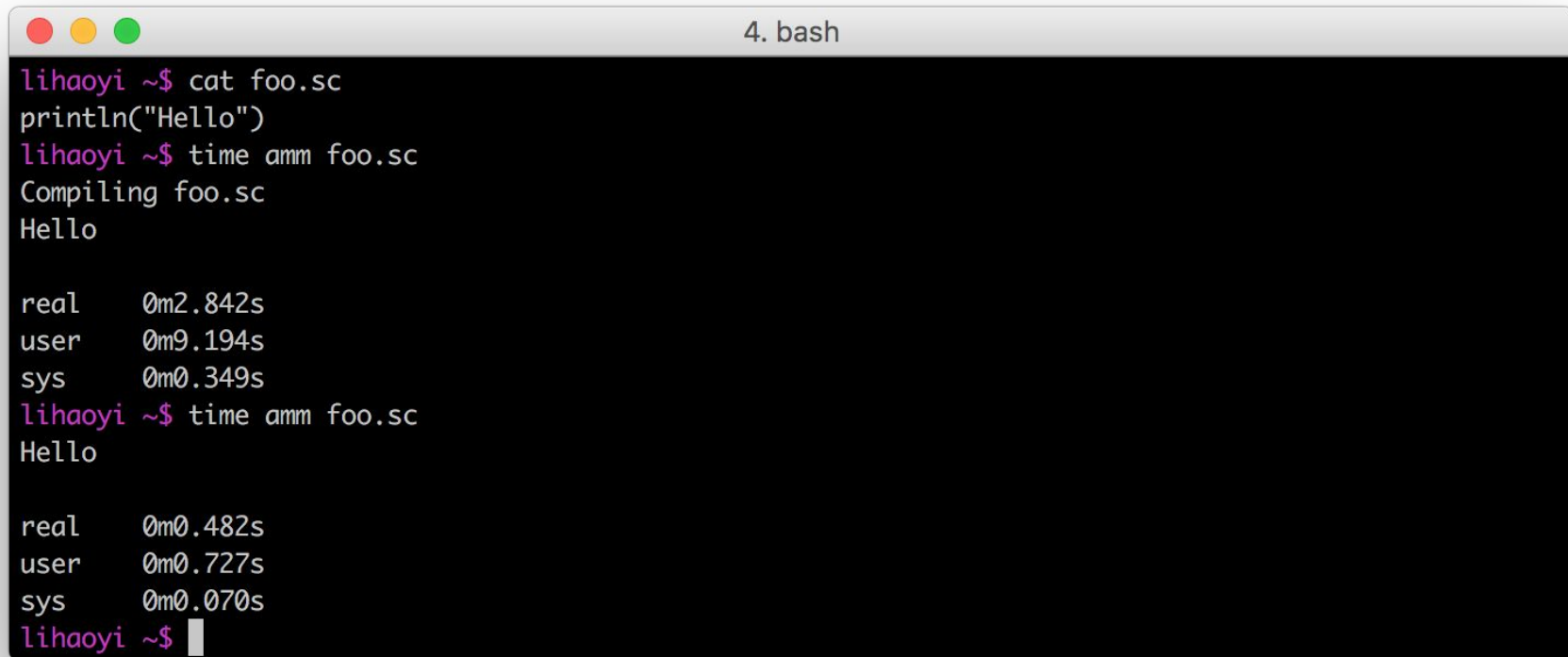
Like the Scala REPL, but better



```
4. java
@ sys.env
res0: Map[String, String] = Map(
  "SHELL" -> "/bin/bash",
  "LOGNAME" -> "lihaoyi",
  "PATH" -> "/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin",
  "TERM" -> "xterm-256color",
  "XPC_SERVICE_NAME" -> "0",
  "TERM_PROGRAM" -> "iTerm.app",
  "HOME" -> "/Users/lihaoyi",
  "ITERM_SESSION_ID" -> "w3t0p1:C608B802-E1A7-4411-BFE4-85AD5F97BB38",
  "Apple_PubSub_Socket_Render" -> "/private/tmp/com.apple.launchd.3516aCJ5Lk/Render",
  "COMMAND_MODE" -> "unix2003",
  "USER" -> "lihaoyi",
  ...
@
```

Ammonite-Scripts

Write Scala without the fuss



```
4. bash
lihaoyi ~$ cat foo.sc
println("Hello")
lihaoyi ~$ time amm foo.sc
Compiling foo.sc
Hello

real    0m2.842s
user    0m9.194s
sys     0m0.349s
lihaoyi ~$ time amm foo.sc
Hello

real    0m0.482s
user    0m0.727s
sys     0m0.070s
lihaoyi ~$
```

Demo: Ammonite REPL

Scala has a code-size gap

Scala REPL: lightweight, works great for < 1 line programs, more is problematic

Ammonite REPL: lightweight, works great for 1 - 10 line programs

SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts

Scala has a code-size gap

Scala REPL: lightweight, works great for < 1 line programs, more is problematic

Ammonite REPL: lightweight, works great for 1 - 10 line programs

Ammonite Scripts: works for 10 - 1,000 line programs

SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts

Demo: Ammonite Scripts

Like running scripts with the `scala` command but...

Can you use third party libraries?

-

Like running scripts with the `scala` command but...

Can you use third party libraries?

-

Can you import other scripts?

-

Like running scripts with the `scala` command but...

Can you use third party libraries?

- Download jar yourself, download transitive dependencies, use `-cp`, ...

Can you import other scripts?

- No

Like running scripts with the `scala` command but...

Can you use third party libraries?

- `import $ivy.`com.lihaoyi::scalatags:0.6.2``

Can you import other scripts?

- `import $file.foo`
- `import $file.foo.bar`

Like running scripts with the `scala` command but...

Can you use third party libraries?

- `import $ivy.`com.lihaoyi::scalatags:0.6.2``
- `libraryDependencies += "com.lihaoyi" %% "scalatags" % "0.6.2"`

Can you import other scripts?

- `import $file.foo`
- `import $file.foo.bar`

Like running scripts with the `scala` command but...

Can you use third party libraries?

- `import $ivy.`com.lihaoyi::scalatags:0.6.2``
- `libraryDependencies += "com.lihaoyi" %% "scalatags" % "0.6.2"`

Can you import other scripts?

- `import $file.foo // loads foo.sc`
- `import $file.foo.bar // loads foo/bar.sc`

Demo: Ammonite Script Imports

Demo: Ammonite Website

Scala has a code-size gap

Scala REPL: lightweight, works great for < 1 line programs, more is problematic

Ammonite REPL: lightweight, works great for 1 - 10 line programs

Ammonite Scripts: works for 10 - 1,000 line programs

SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts

Use Cases

www.lihaoyi.com

www.lihaoyi.com/Resume

github.com/lihaoyi/Ammonite/blob/master/ci/build.sc

Next Steps

Next Steps: need more usage

- Currently about 6-7k downloads a month
- More users = more edge cases get found, more help fixing bugs,
- More users = more info to help set direction of development

Next Steps: need more contributors!

- Lots of people contributing fixes, but still mostly me driving development
- Script-runner was mostly built by two GSOC interns, but they've moved on
- Need more people to sustainably drive project forward

Next Steps: concrete tasks

- IntelliJ support for scripts
- Convert autocomplete to use 2.12 REPL's improved autocomplete
- Experiment with using shapeless for more stable pretty-printing
- Better SBT integration: re-use compiler settings? Plugins?
- Spark support??

Scala has a code-size gap

Scala REPL: lightweight, works great for < 1 line programs, more is problematic



?

SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts

Scala has a code-size gap

Scala REPL: lightweight, works great for < 1 line programs, more is problematic

Ammonite REPL: lightweight, works great for 1 - 10 line programs

Ammonite Scripts: works for 10 - 1,000 line programs

SBT: Works great for 1,000 - 100,000 line programs, overkill for small scripts

Scala Scripting

Scala By the Bay, San Francisco, 12 Nov 2016

Q & A